109

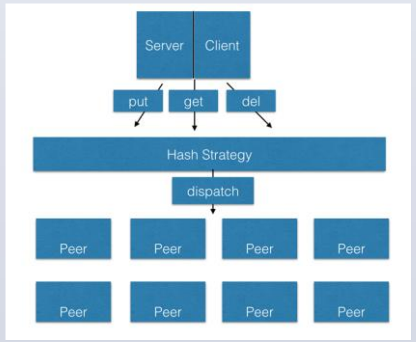
Distributed Key-Value Store on Amazon EC2 Cloud System

Qinyao XU A20304397

Abstract

My design is a distributed hash table system. It has no central server, each peer is both a server and a client. As a client, it provides interfaces through which users can issue queries and view search results. As a server, it accepts queries from other peers, checks for matches against its local hash table, and responds with corresponding results. In addition, since there's no central indexing server, search is done through consistent hashing. In this evaluation, it tells it works not bad on cloud environment from latency and throughput.

Architecture and Design

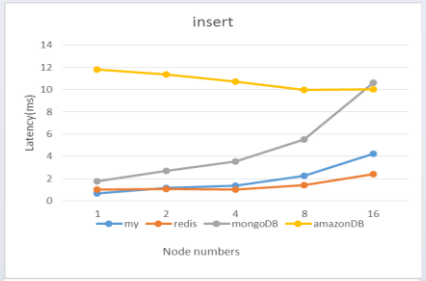


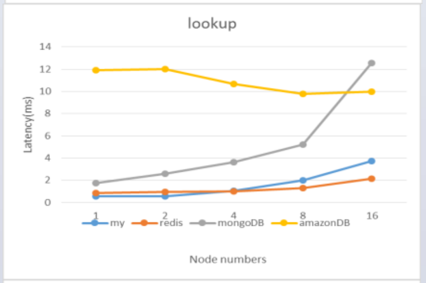
Experiment setup

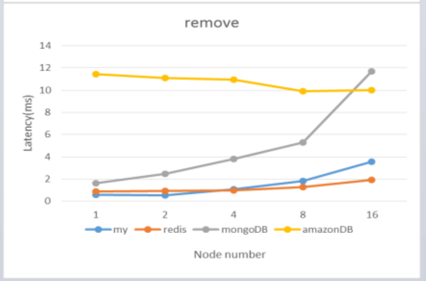
Commercial Cloud
Amazon EC2
M3.medium
Commodity Cluster
Up to 16 nodes

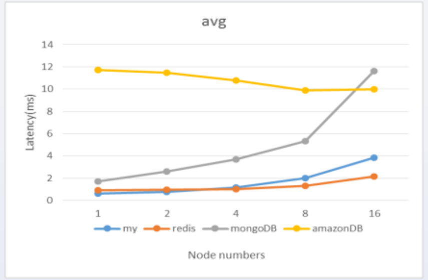
Micro benchmark settings
N clients to N servers: All-to-All communication pattern
Random generated key–value pairs: 10bytes key, 90 bytes value

Latency

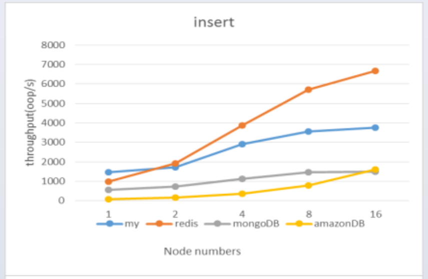


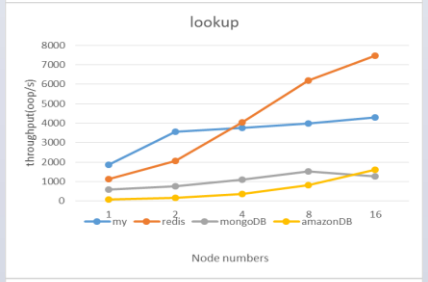


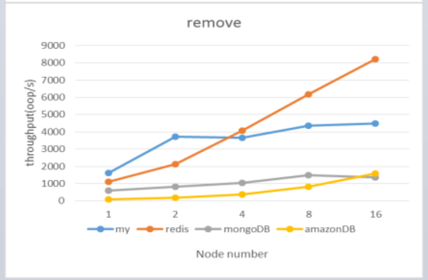


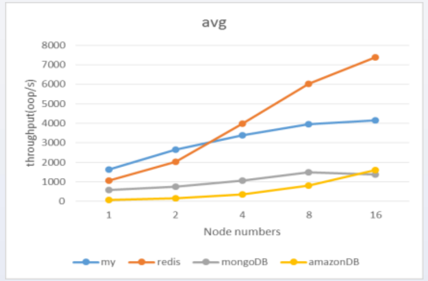


Throughput









Conclusion

According to the above picture it shows that my design has a good performance and scalability, also we find out that Redis has the best performance through all the four system, it has a latency less than 2ms even in 16 nodes. And MongoDB has a bad scalability, the latency will increase very fast when the nodes numbers increase. DynamoDB didn't change a lot when the number of nodes increases.

Related Work and References

Tonglin Li, Xiaobing Zhou, Ke Wang, Dongfang Zhao,Iman Sadooghi, Zhao Zhang, Ioan Raicu
A Convergence of Key-Value Storage Systems from Clouds to Supercomputers,Wiley InterScience
Tonglin Li, Xiaobing Zhou, Kevin Brandstatter, et al.
MongoDB,Redis,Amazon DynamoDB, developer ducuments and API